



Themed Common Controls:

**A New Look for Common
Controls on Whistler**

**PM: Gregory Raiz
Dev: Lou Amadio**



Agenda

Overview:

- What are we doing and why?

- Intro to ComCtl32 and themes

Using new ComCtl32 V6 in your program

- Applications, DLL's and control panels

- Testing and common problems

Using themes on uncommon controls

Fusion details and other scenarios

New features in comctl32

Questions and answers



What Are We Doing?

Providing a cleaner, simpler visual style for business windows

Providing a second inviting and exciting visual style for consumer windows

Providing mechanisms for ISVs to synch with these new styles



What Are the Benefits?

Huge opportunity to provide users with an improved experience

Visual styles will be honed for our different user types

In most cases, development cost is minimal

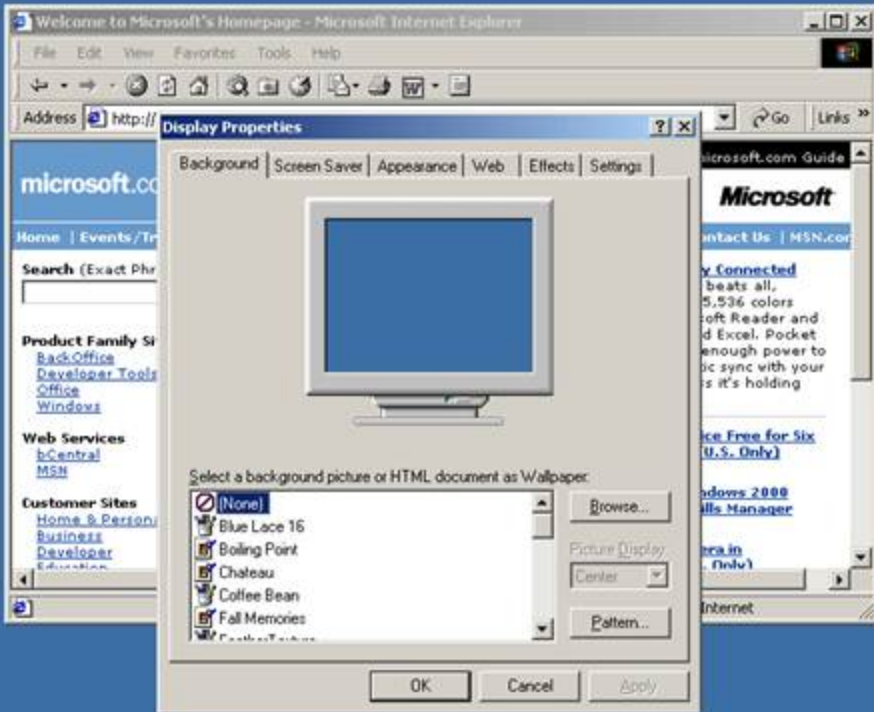
This work allows your applications to automatically receive future visual refinements

Generate consumer excitement

Let's take a look...



Recycle Bin



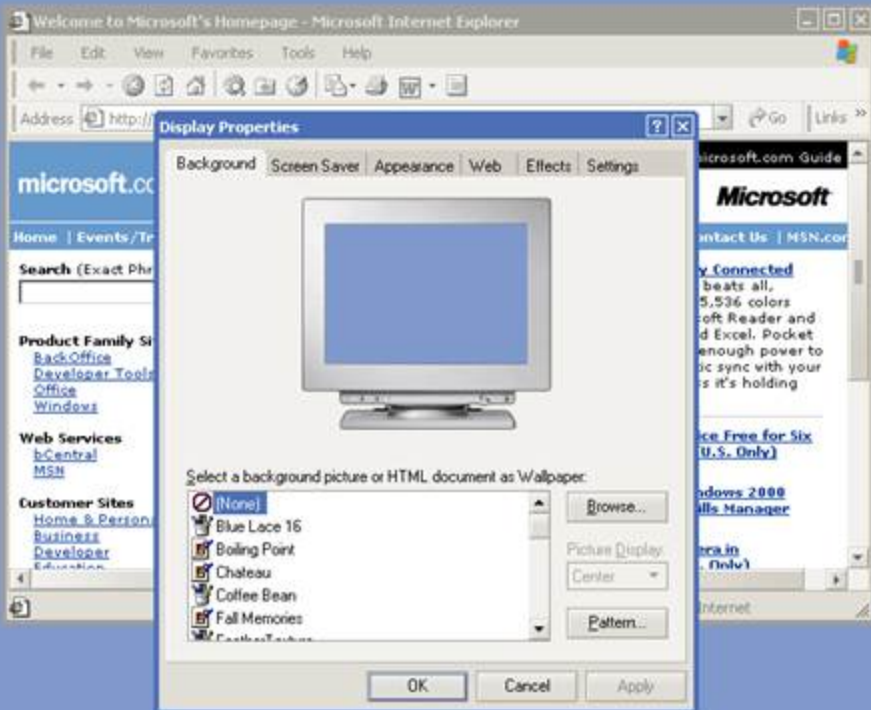
Start

Welcome to Microsoft's H...

9:55 PM



Recycle Bin



Start



Welcome to Microsoft's H...



9:55 PM



What's Possible?

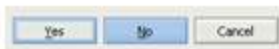
Non-rectangular
controls



Alpha Support



Hot state for most
controls





User Experience Theme Manager (uxtheme.dll)

- Provides rendering support for the new controls

- Provides information APIs to assist controls in positioning parts

- Renders new visual style for the non-client area of all applications

 - Unless they are custom drawn

 - Includes scroll bars and Title bars



Common Controls (Comctl32.dll V6)

We have coalesced our user win32 controls with our shell common controls in Commctl v.6

These new controls use the theme manager to render appropriate visuals

Existing applications will continue to use Comctl v.5.8 unless they provide a manifest

Comctl v.6 will not ship down-level



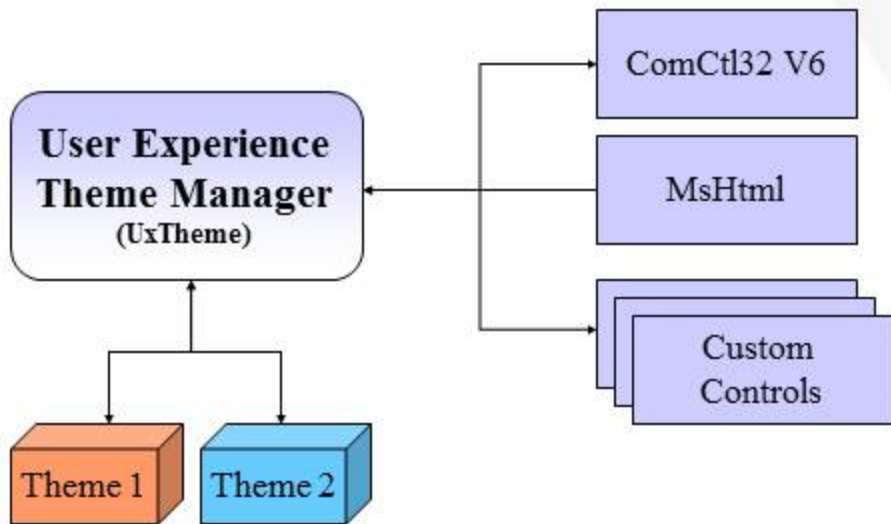
Theme Manager + ComCtrl

Allows us to theme the shell and applications that use the new comctl32

Visual flexibility gives designers the ability to innovate the windows look

Open API's allow ISV's to easily get the new look in their application

How Does This Work?





Why a New Common Control?

Top goal: Don't break existing applications

Second Goal: Create consumer excitement

Goals are in conflict

Changing anything in Comctl32 breaks applications.



Why a new Common Control? Cont.

Used Fusion to meet both goals
Old and New Comctl run side by side
Future versions won't break your
application
Applications get the version that
they tested



Why a theme manager?

Needed refresh the look of Windows

Many areas that needed to be refreshed

Needed to allow designers to innovate
independent of the development process

Let's see it in action...



Business Style

Simple Simple Simple - by design
Similar to Office10 - by design
Tuned for professional workers



How Do You Add Support?

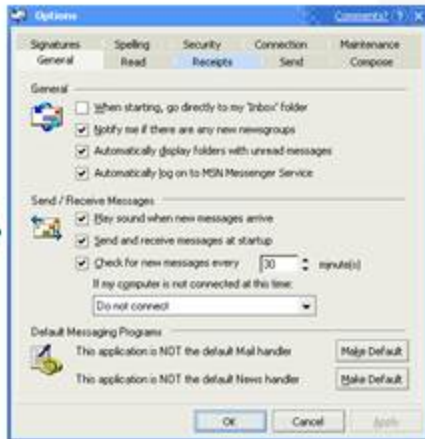
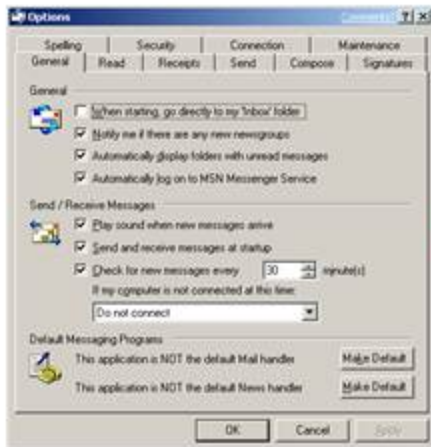
Use the new Common Controls by adding a fusion manifest.

For your custom controls, the theme manager provides rendering API's

Demo

whistler

Theming: Outlook Express





Testing One, Two, Three ...

Comctl32 v.6 has added more parameter validation

Additionally the underlying structure of many controls has been improved for stability

This requires that you perform a thorough test pass on your application before releasing a manifest



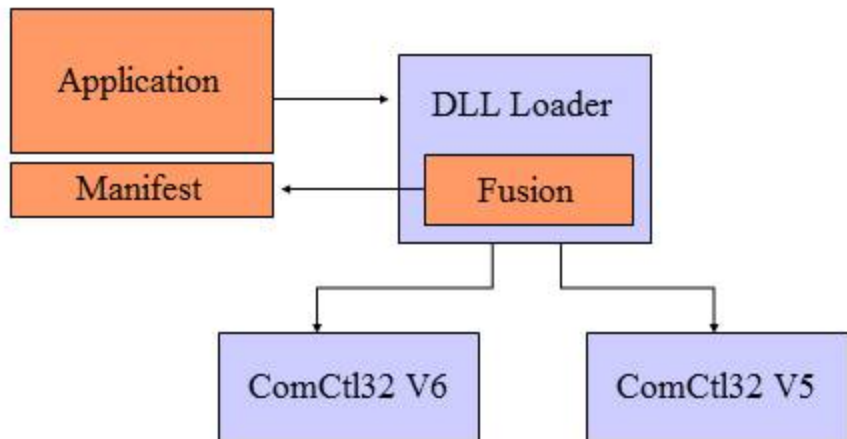
Using the new Common Controls

Use a fusion manifest to load the common control assembly

A sample Fusion manifest:

```
<Assembly manifestversion="1.0"  
  name="OutlookExpress" description="email">  
  <dependency assemblyname="comctl32"  
    version="6.0.0.0" />  
</Assembly>
```

How does Fusion Work?





Theming a Simple Application or Control Panel applet

Create a manifest

"YourAppName.exe.manifest"

```
<assembly manifestversion="1.0" name="Your App Name"  
    version="1.0.0.0" description="Your App's description">  
    <dependency assemblyname="comctl32" version="6.0.0.0" />  
</assembly>
```

Executable Example:

c:\Program Files\Outlook Express\msimn.exe.manifest

Control Panel Example:

c:\windows\system32\desk.cpl.manifest



Theming a Simple Application, cont.

Warning: Do not use this method if 3rd party extensions can get pulled into your process. You must test your application.

We have removed app hacks that may have kept your application working, if your app breaks please fix it.



Theming a Simple Application, cont.

1. Link to comctl32.lib, InitCommonControls(Ex)
2. Add a file that has the manifest format.
3. Add an entry to your resource (.rc) file that points to the manifest:
1 RT_MANIFEST "YourApp.manifest"
4. There is no step 4. Your app is fusionized



Theming an application with 3rd party extensions. (PhotoShop, MMC, Explorer)

To ensure compatibility you need to control what extensions are using ComCtl32 V6.

This involves managing the application manifest manually

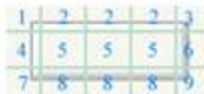
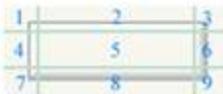
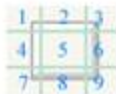
For more information, visit:

<http://shell/development/documents/whistler/fusionstart.asp>



What UxTheme does under the covers

Internally we use 9 grid bitmap scaling



Rendering is transparent to Comctl32. In future versions we can use vectors or other rendering techniques.

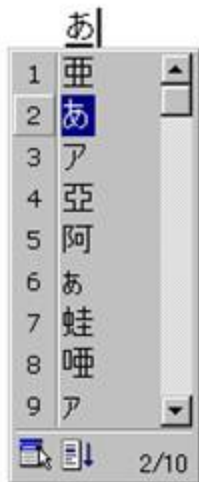
These future enhancements can be made without changing the controls.

Control are visually consistent across architectures (html, win32, other)

Theming UnCommon Controls

Separate the control into states and parts.

Before:



After:



What is a Part?

Toolbar has the following parts:



TP_BUTTON:  Search

TP_DROPDOWN: 

TP_SPLITBUTTON:  Back

TP_SEPARATOR: |

TP_SPLITBUTTONDROPDOWN: 

“Part 0” (The unnamed part):



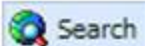
What is a state?

Toolbar's button has states:

TS_NORMAL:



TS_HOT:



TS_PRESSED:



TS_CHECKED:



TS_DISABLED:





Code Sample for custom themeing

Get theme handle in WM_CREATE

```
hTheme = OpenThemeData(hwnd, L"Toolbar");
```

Draw using Theme APIs

```
HRESULT hr = E_FAIL;
```

```
if (hTheme)
```

```
    hr = DrawThemeBackground(hTheme, hdc, TP_BUTTON,  
                             TS_NORMAL, &rc, 0);
```

```
if (FAILED(hr))
```

```
    // If we don't have a theme handle or if there was an error during  
    // the call to the theme manager, draw classic style....
```

Close the theme handle in WM_DESTROY

```
CloseThemeData(hTheme);
```



What Happens on Down Level Platforms?

ComCtl32 v6 will not ship down level.

On a down level platform continue to use ComCtl32 version 5.81.

Version 5.81 does not have: new visuals, hyperlink control, or group view. If you use these new features you should test your application on Win95, Win98 and Win2000.



New Features

Grouping

Tile view

Faded Toolbars

Hyperlink



Grouping

Allows you to
group item by
their
properties

Easier to read
and parse long
lists

Extensible
with other
properties

File Folder (4 items)



Remote
Desktops



My Videos



My Pictures



Greg Ratz's
Music

HTML Document (9 items)



test_select...



Stock Services



scope
scenarios



Microsoft
401(k)



layout_draft

Internet Shortcut (2 items)



richTextEditor



Photo Gallery

Shell Favorite Folder



Favorites

Tile View

New View, icon with
two-three lines of
information

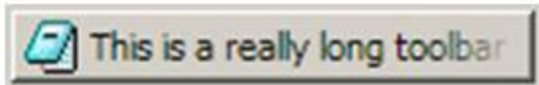
Useful for
presenting more
info then just the
name of the item





Faded Toolbars

Provides three more letters rather than an ellipsis



Hyperlink

Control for Win32 applications

Text with blue underline that can navigate to a web page



Changes to User controls

BCM_GETIDEALSIZE - Used to ask button for it's size.

BCM_SETIMAGELIST - Used to allow a control container to set an image next to text in the control. Used to theme the Start Button

Flicker Free optimizations.



Caveat for New Features

When using a new feature fail gracefully

New features are Whistler only

If you ship down level test it with V5.8

Comctl and test it down level



Summary

New themed controls

Add a manifest to avoid dll hell

Test, test, test



Questions

Additional information

SDK

<http://zanzibar/shellccdocs/userexframes.htm>

Fusion <http://fusion>

Slide deck:

<http://windowsclient/public/comctl32.ppt>

List of parts and states:

`\\zanzibar\ShellCCDocs\UserEx\reference\topics\PartsAndStates.xml`

ComCtrl@microsoft.com

DllHell@microsoft.com